

Agile Software Development

Overview

What is “Agile”

- **“Agile is an iterative, incremental and evolutionary approach to software development which is performed in a highly collaborative manner by self-organizing teams with "just enough" ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.”**

Scott Ambler, Agile Practice Leader, IBM

Iterative

- Team implements requirements in small steps called Iterations
- Requirements chosen are the highest priority at that time (from the business perspective)
- Requirements are fully implemented, tested and accepted within an Iteration – ***“Done, Done”***

Incremental

- System is released early and often
- Mitigates risk of business change
- Delivers value and thus ROI sooner
- Focuses the team on delivery rather than process

Evolutionary

- Constantly delivering an *acceptably incomplete* system, i.e. not waiting for “perfect”
- Does not mean low quality hacked solutions, but managed scope
- Allows real-world feedback to be factored in to the working system

Highly Collaborative

- Team should be co-located
 - Exponential increase in communication
 - 50% performance improvement
- Team consists of:
 - Business/Stakeholder representatives, Subject Matter Experts, QA, Development, Data Management, Enterprise Architecture, Infrastructure, IT Finance, Project Management Office, Facilities, HR, ...
- Release and Iteration Planning, daily *Huddle*, ad hoc meetings as required, Retrospectives

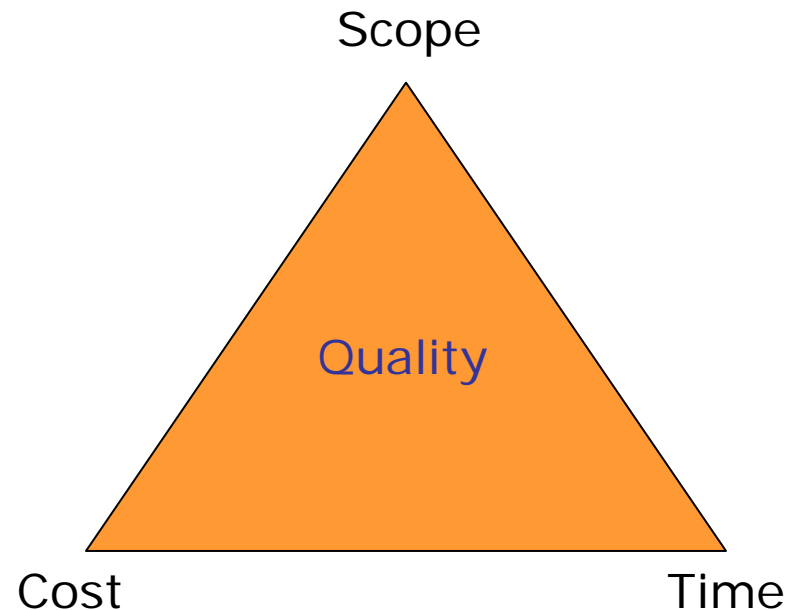
Self-organizing Teams

- Team accountable to the Stakeholders, i.e. deliver running, tested features regularly
- Stakeholders accountable to the Team, i.e. own and manage the requirements
- Pays off with understanding, quality and commitment
- Project Management role is different – facilitator of communication, *servant leadership*
- Requires constant communication and respect

Just Enough “Ceremony”

- Analogous to electrical system in a car
- Larger the team and/or organization, the more ceremony required
- Easier to start small and grow rather than start large and reduce
- Value Stream Mapping from Lean
 - Ruthlessly eliminate unnecessary waste

High Quality



- Quality is non-negotiable, and always “high”; Scope varies instead
- Quality through testing; features must be “**Done, Done**”
- Quality through feedback; Iterative and Incremental delivery ensures features meet business needs

Cost Effective

- Lower up-front investment
- Faster ROI through incremental release
- Reduced waste on features rarely used
- Process visibility identifies problems sooner
- Can “cut and run” if required, and will still have delivered value

Meets Changing Needs

- Accommodates business changes
 - Incremental delivery reduces risk
 - Planning after each Iteration handles change
- Incorporates usage feedback
 - Small release cycle allows key issues to be resolved
 - Accommodates new ideas from “real-world”

Simple Communication

Project Dashboard – Snapshot of Current Status

Big Visible Chart
LVC-Process Performance Dashboard

ITERATION 1	
Story Points this Iteration	10
Points Remaining this Iteration	3 ☺
Total Story Points	66
Total Points Remaining	59
Average Velocity	10
Outstanding Defects	5

Simple Communication

Release and Iteration Plans



“Brands”

- Lean Software Development
- Extreme Programming (XP)
- Industrial XP
- Scrum
- Crystal Clear
- Feature Driven Development (FDD)
- Adaptive Software Development
- Dynamic System Development Method (DSDM)